

Possibilistic C-Shell Clustering with Inter-Cluster Constraints

Tsaipei Wang, *Member, IEEE*, and James M. Keller, *Fellow, IEEE*

Abstract—This paper describes our analysis of using extra constraint terms regarding relations between cluster prototypes in possibilistic c-shell clustering. The extra constraints are implemented as additional terms in the cost function. This allows users of these algorithms to incorporate additional knowledge regarding properties cluster prototype into the clustering process. Our analysis here focuses on the use of one extra term for locating circles (shell clustering with circular prototypes) with similar radii. An adjustable factor is used to control the strength of this constraint. For possibilistic clustering, this couples the update procedure of the otherwise independent prototypes. Our experiments, using both simulation and real image data, indicate that this is especially useful in locating actual clusters when the available data are noisy.

I. INTRODUCTION

THE ability to locate specific shapes in images is important for many image analysis problems. Fuzzy and possibilistic c-shell clustering algorithms have been demonstrated to perform well for the task of locating lines, circles, and quadric curves in 2-D images (or hyperplanes, hyperspheres, and hyperquadric surfaces in higher dimension data) [1-4], rectangles [5], and template-based shapes [6]. Examples of their application to real-world image analysis are in [7,8]. Among the two (fuzzy and possibilistic) approaches, the possibilistic approach is more immune to the effect of noise points, and is the basis of our algorithm discussed in this paper. Compared with the generalized Hough transform [9] or template matching, shell clustering has much smaller memory and time requirements, and the precision of prototype parameters is not limited to the bin size in the Hough transform or the parameter resolution of the templates [2].

Existing c-shell clustering algorithms allow the specification of the number and type of prototypes, the initial conditions (membership values or prototypes), and a distance or similarity measure between prototypes and data points. They do not allow the inclusion of additional constraints on cluster characteristics in the clustering process. However, in some particular applications there may be other desirable properties for the clustering results. For these cases, we present here the idea of introducing approximate constraints on cluster prototype parameters that correspond to the desired cluster properties. Since fuzzy and possibilistic

clustering algorithms identify the clusters through minimization of cost functions, it seems natural to try to incorporate these additional constraints on cluster parameters in the cost functions, and this is the approach we describe in this paper.

There are many different types of additional constraints, depending on the particular applications and the extra information we know about the desired clusters. Such constraints can also be applied to the relations between clusters. For example, in c-shell clustering, we may want linear/planar prototypes that are approximately parallel to each other, or circular prototypes with similar radii. Different constraints mean different constraint terms added to the cost function. Here we focus on one example, where we look for circular prototypes that have similar radii, and use this example to illustrate some properties of using such additional constraints. In the following, section II describes the formulation of the original and our modified possibilistic c-shell clustering (PCSC) algorithms for circles. Section III contains simulation results, and section IV describes an application in a real world scenario. Section V is our conclusion and discussion.

II. DESCRIPTION OF ALGORITHM

A. The Original PCSC for Circles

The original PCSC algorithm, presented in [2], is an extension of the possibilistic c-means (PCM) algorithm [10] with shell-shaped prototypes. Here we limit our description to circle prototypes only. The following distance measure between a point $\mathbf{x}_j$ and a circle prototype $\{c_i, r_i\}$,

$$d_{ij}^2 = \left(\|\mathbf{x}_j - c_i\|^2 - r_i^2 \right)^2, \quad (1)$$

is used to allow an analytical form for updating the prototypes, avoiding the numerical methods used by [4] for the original fuzzy c-shell clustering. The objective function of PCM is

$$J = \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m d_{ij}^2 + \sum_{i=1}^C \eta_i \sum_{j=1}^N (1 - u_{ij})^m. \quad (2)$$

Here u_{ij} is the possibilistic membership of vector $\mathbf{x}_j$ in prototype $\{c_i, r_i\}$, N is the number of vectors, C is the number of clusters, and η_i and m are two parameters that determine the dependence of u_{ij} on d_{ij} . The update equations for the possibilistic membership values and the prototypes are

Tsaipei Wang is with the Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan (phone: +886-3-5712121 ext.56689; e-mail: wangts@cs.nctu.edu.tw).

J. M. Keller is with the Department of Electrical and Computer Engineering, University of Missouri-Columbia, Columbia, MO 65211, USA.

$$u_{ij} = \left[1 + \left(\frac{d_{ij}^2}{\eta_i} \right)^{\frac{1}{m-1}} \right]^{-1} \quad (3)$$

and

$$\mathbf{p}_i = -\frac{1}{2}(\mathbf{H}_i)^{-1} \mathbf{w}_i, \quad (4)$$

respectively, where

$$\mathbf{p}_i = \begin{bmatrix} -2c_i \\ \mathbf{c}_i^T \mathbf{c}_i - r_i^2 \end{bmatrix}, \quad \mathbf{H}_i = \sum_{j=1}^N u_{ij}^m \begin{bmatrix} \mathbf{x}_j \\ 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_j^T & 1 \end{bmatrix},$$

$$\mathbf{w}_i = 2 \sum_{j=1}^N u_{ij}^m (\mathbf{x}_j^T \mathbf{x}_j) \begin{bmatrix} \mathbf{x}_j \\ 1 \end{bmatrix}, \quad \text{and} \quad b_i = \sum_{j=1}^N u_{ij}^m (\mathbf{x}_j^T \mathbf{x}_j)^2.$$

These update equations are derived by setting to zero the derivatives of the objective function J relative to u_{ij} and $\mathbf{p}_i$.

We can perform the original PCSC algorithm using the procedure below:

1. Initialize the prototypes.
2. Repeat until convergence or for a predefined number of iterations
 - 2.1. Calculate cluster membership $\mathbf{U} = \{u_{ij}\}$ using (3).
 - 2.2. Update the prototypes using (4).

Convergence can be defined as when the changes of prototypes between iterations fall below a preset threshold.

B. The Modified PCSC for Circles

Since we are considering circle prototypes with similar radii, we add an extra term J_c to the original cost function J in (2) to take radius similarity into account:

$$J' = J + J_c. \quad (5)$$

For the purpose of simplicity, let us consider only two clusters now. Let (c_1, r_1) and (c_2, r_2) represent the respective prototypes. For the constraint of radius similarity, we can use the following form for J_c :

$$J_c = \alpha (r_1^2 - r_2^2)^2. \quad (6)$$

Here α is zero or a positive number indicating how strong the constraint is. The selection of its value will be discussed later.

The next step is to derive the update equations. With the added term J_c , we are not able to derive a single equation like (4) that simultaneously updates both the center and radius of a prototype. It is necessary to differentiate J' relative to the center and radius of a prototype separately, and minimize J' by setting these derivatives to zero:

$$\frac{\partial J'}{\partial \mathbf{c}_i} = -2 \sum_{j=1}^N (u_{ij})^m (\mathbf{x}_j - \mathbf{c}_i) \left(\|\mathbf{x}_j - \mathbf{c}_i\|^2 - r_i^2 \right) = 0, \quad (7)$$

and

$$\frac{\partial J'}{\partial (r_i^2)} = -2 \sum_{j=1}^{N_i} (u_{ij})^m \left(\|\mathbf{x}_j - \mathbf{c}_i\|^2 - r_i^2 \right) + 2\alpha (r_i^2 - r_{3-i}^2) = 0. \quad (8)$$

Here $i=1, 2$.

Equations (7) and (8) do not allow us to obtain simple closed-form update equations for the prototype parameters, $\mathbf{c}_1$, r_1 , $\mathbf{c}_2$ and r_2 . This is because both r_1 and r_2 appear together due to the added term J_c . Therefore, an iterative method is necessary for finding the solutions. Similar to [4], we choose to solve (7) and (8) using Newton's method. There are 6 quantities to be optimized simultaneously (2 each for $\mathbf{c}_1$ and $\mathbf{c}_2$ for two-dimensional data, and one each for r_1 and r_2).

The procedure and stopping criteria of the modified PCSC algorithm are similar to those of the original PCSC, except that we update the prototypes by solving (7) and (8) using Newton's method. To reduce the amount of computation time, we further limit the prototype parameter updating step to only one iteration of Newton's method. This was indicated in [11] to still give good clustering results, which we find to be the case.

In order to gain some insight into the effect of the new parameter α , let us first consider a "perfect" scenario where the data points fall exactly on two circles and are evenly spaced in each circle, with $(\mathbf{c}_1^*, r_1^*)$ and $(\mathbf{c}_2^*, r_2^*)$ being the centers and radii of the two underlying circles. In addition, these two circles are assumed to be well separated so their corresponding data points can be treated as two separate sets (called $\mathbf{X}_{(1)}$ and $\mathbf{X}_{(2)}$ below), with the membership of a data point in one set to the cluster of the other set assumed to be zero. This makes the clustering process more like two separate clustering processes of $\mathbf{X}_{(1)}$ and $\mathbf{X}_{(2)}$ coupled only by the constraint term J_c .

For this special case, one solution to (7) is when $\mathbf{c}_i = \mathbf{c}_i^*$, regardless of the value of r_i . This can be understood by considering the assumed symmetry. When $\mathbf{c}_i = \mathbf{c}_i^*$, all the $\mathbf{x}_j$ in $\mathbf{X}_{(i)}$ have the same d_{ij} :

$$d_{ij}^2 = \left[(r_i^*)^2 - r_i^2 \right]^2 \quad \text{for } \mathbf{x}_j \text{ in } \mathbf{X}_{(i)},$$

which is independent of j . This makes the memberships u_{ij} independent of j as well. (We will use $u_{(1)}$ and $u_{(2)}$ to represent the j -independent memberships.) Combine these with the assumption of evenly spaced $\mathbf{x}_j$, and we can see that the summation in (7) becomes zero when $\mathbf{c}_i = \mathbf{c}_i^*$. Since we have not made any assumption regarding the prototype radii, this proves that $\mathbf{c}_i = \mathbf{c}_i^*$ is a solution.

Our next task is to solve for r_1 and r_2 given $\mathbf{c}_1 = \mathbf{c}_1^*$ and $\mathbf{c}_2 = \mathbf{c}_2^*$. First, from (8) we can obtain the following relation between r_1 and the other parameters:

$$r_1^2 = \frac{\alpha r_2^2 + \sum_{\mathbf{x}_j \in \mathbf{X}_{(1)}} (u_{(1)})^m \|\mathbf{x}_j - \mathbf{c}_1\|^2}{\alpha + \sum_{\mathbf{x}_j \in \mathbf{X}_{(1)}} (u_{(1)})^m}. \quad (9)$$

Let N_1 and N_2 are the number of data points in $\mathbf{X}_{(1)}$ and $\mathbf{X}_{(2)}$, respectively, and $N_1' = N_1 (u_{(1)})^m$ and $N_2' = N_2 (u_{(2)})^m$ be the total membership in the two individual clusters. By into (9) substituting $\|\mathbf{x}_j - \mathbf{c}_1\| = \|\mathbf{x}_j - \mathbf{c}_1^*\| = r_1^*$ for $\mathbf{x}_j$ in $\mathbf{X}_{(1)}$, we get

$$r_1^2 = \frac{\alpha r_2^2 + N_1'(r_1^*)^2}{\alpha + N_1'} \quad (10a)$$

and similarly

$$r_2^2 = \frac{\alpha r_1^2 + N_2'(r_2^*)^2}{\alpha + N_2'} \quad (10b)$$

We can further derive separate expressions $(r_1)^2$ and $(r_2)^2$ as weighted averages of $(r_1^*)^2$ and $(r_2^*)^2$:

$$r_1^2 = \frac{(\alpha N_2')(r_2^*)^2 + (\alpha + N_2')N_1'(r_1^*)^2}{(\alpha N_2') + (\alpha + N_2')N_1'} \quad (11a)$$

and

$$r_2^2 = \frac{(\alpha N_1')(r_1^*)^2 + (\alpha + N_1')N_2'(r_2^*)^2}{(\alpha N_1') + (\alpha + N_1')N_2'} \quad (11b)$$

This implies that solutions of both r_1 and r_2 have values between r_1^* and r_2^* . When $\alpha=0$, we have $r_1=r_1^*$ and $r_2=r_2^*$ as expected. When α is much larger than N_1 and N_2 (and N_1' and N_2' as well), it turns out that r_1 and r_2 become identical:

$$r_1^2, r_2^2 \approx \frac{N_1'(r_1^*)^2 + N_2'(r_2^*)^2}{N_1' + N_2'} \quad (12)$$

Equations (11) and (12) are not solutions of r_1 and r_2 . This is because N_i' actually depends on r_i through d_{ij}^2 and $u_{(i)}$. A special case when we can obtain simple analytical solutions for r_1 and r_2 is when we set $\eta_1, \eta_2 \rightarrow \infty$, and from (3) we have $u_{ij}=1$. This in turn makes $N_i'=N_i$. This allows us to compute analytical solutions of r_1 and r_2 directly using (11), with N_1' and N_2' replaced by N_1 and N_2 . In addition, because membership values are constant, these solutions will not change between iterations of the clustering algorithm. In other words, they correspond to the prototypes found by our modified PCSC algorithm under the conditions above. In the following we continue to assume $N_i'=N_i$.

The purpose of the derivations above is to understand how the relations between r_1 , r_2 , r_1^* and r_2^* are affected by the relations between α , N_1 and N_2 . One interesting aspect to look at is the ratio between $[(r_1)^2 - (r_1^*)^2]$ and $[(r_2)^2 - (r_1^*)^2]$. This is a measure of how much the radius of a prototype is affected by the difference between the two underlying circles. We can easily obtain this ratio by subtract $(r_1^*)^2$ from both sides of (11a):

$$\frac{(r_1)^2 - (r_1^*)^2}{(r_2^*)^2 - (r_1^*)^2} = \frac{\alpha N_2}{\alpha N_2 + (\alpha + N_2)N_1} \quad (13a)$$

and similarly,

$$\frac{(r_2)^2 - (r_2^*)^2}{(r_1^*)^2 - (r_2^*)^2} = \frac{\alpha N_1}{\alpha N_1 + (\alpha + N_1)N_2} \quad (13b)$$

We can draw a few observations for special cases of (13):

(1) The ratio in (13a) is close to one when $\alpha N_2 \gg (\alpha + N_2)N_1$ (i.e., after divided by $\alpha N_1 N_2$, $N_1^{-1} \gg \alpha^{-1} + N_2^{-1}$). This occurs when both α and N_2 are much larger than N_1 . In this case, we have $r_1 \approx r_2^*$. On the other hand, the ratio in (13b) is close to zero under the same condition because $\alpha N_1 \ll \alpha N_2 < (\alpha + N_1)N_2$, resulting in $r_2 \approx r_2^*$. One observation we can draw

is that, when α is large compared with the number of data points, and there are many more data points in one set than the other, both prototypes will have radius similar to that of the underlying circle of the larger data set.

(2) When $\alpha \gg N_2 = N_1$, both ratios in (13) become 1/2, giving $r_1^2 \approx r_2^2 \approx [(r_1^*)^2 + (r_2^*)^2]/2$. This means that both prototypes end up with the same radius when α is very large and both sets have the same number of data points.

(3) The ratio in (13a) is close to zero when $\alpha N_2 \ll (\alpha + N_2)N_1$ (i.e., after divided by $\alpha N_1 N_2$, $N_1^{-1} \ll \alpha^{-1} + N_2^{-1}$). This occurs when either α or N_2 is much smaller than N_1 . In this case, we have $r_1 \approx r_1^*$, meaning that r_1 is barely affected by the extra constraint.

Overall, for the extra constraint to significantly affect the clustering result, α should be large compared with either N_2 or N_1 .

For more thorough investigation of how the relations between α , N_1 and N_2 affect the clustering result, we plot in Fig. 1 r_1 (solid lines) and r_2 (dashed lines) relative to r_1^* and r_2^* as functions of α for a few different combinations of N_1 and N_2 , computed from (11). The radii of underlying circles in this simulation are set to $r_1^*=10$ and $r_2^*=20$. We can see that $r_1=r_1^*$ and $r_2=r_2^*$ when $\alpha=0$, and that both r_1 and r_2 appear to move toward the same value when α is increased. In addition, this value shifts toward r_1^* when N_1/N_2 increases, and toward r_2^* when N_1/N_2 decreases.

III. SIMULATION RESULTS

In this section our modified PCSC algorithm is applied to two dimensional simulation data. The assumption of independent sets of data for individual clusters is used throughout this section just for illustration purpose. This assumption does not affect the actual usage of the algorithm because in the possibilistic approach to clustering, each prototype is updated independently regardless of what happens with the other prototypes, except for our added inter-cluster constraints.

For comparison with the derivation in the previous section, we start with the "perfect" scenario. We plot in Fig. 1 r_1 (squares) and r_2 (circles) obtained by applying our modified PCSC algorithm, instead of using the analytical results, for a few different values of α . The simulation results are obtained with perfect initialization (initial prototypes overlapping with underlying circles of the data points), $\eta=500000$, and $m=1.5$. The value of η is made very large so that all membership values will be close to one, as assumed for the analytical results. We can see the prototype radii obtained through clustering are consistent with the analytical results.

Fig. 2 contains the data points and prototypes found with various combinations of α , N_1 and N_2 . Parameter values $\eta=500000$ and $m=1.5$ are used. The small dots are data points and the large circles are prototypes found. Fig. 2(a) and (b) have $\alpha=0$, and the prototypes found match perfectly

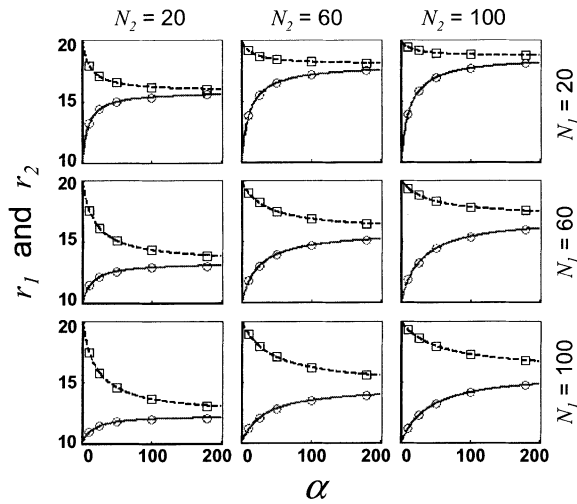


Fig. 1. Radii of prototypes found by PCSC relative to the radii of underlying circles of the data points, as functions of α , N_1 and N_2 . Each individual plot corresponds to a different combination of N_1 and N_2 . Radii of underlying circles are $r_1^*=10$ and $r_2^*=20$. The solid and dashed lines are analytical results for r_1 and r_2 , respectively, and the small squares and circles are simulated results for r_1 and r_2 , respectively.

with the underlying circles of the data points. In Fig. 2(c), we have $\alpha > 0$ and $N_1 \neq N_2$, but both prototypes match perfectly with the underlying circles and have the same radius because we have set $r_1^* = r_2^*$. Fig. 2(d)-(f) correspond to $\alpha > 0$, $N_1 \neq N_2$, and $r_1^* \neq r_2^*$. In Fig. 2(d), we have $N_2 > N_1$, and the resulting prototypes match better with $X_{(2)}$. This is reversed in Fig. 2(e). Fig. 2(f) is the same as Fig. 2(d) except that α is larger, causing the two prototypes to have even closer radii. The change is more evident for $X_{(1)}$, which has less data points.

It may seem strange that the extra constraint term actually makes the prototypes different from the underlying circles with different radii, such as in Fig. 2(e). The analysis using the perfect scenario is to help us understand the relations between α and other parameters. A more practical use of the modified PCSC algorithm is to more accurately locate the prototypes that are expected to have similar radii when the data are noisy. For example, assume that $X_{(1)}$ contains only data points along a circle; we can find this underlying circle with PCSC easily. In the other hand, for example, let $X_{(2)}$ be so noisy that simply clustering its data points produces a prototype very different from the underlying cluster. In such cases, we can think of the "coupled" clustering as having the prototype for $X_{(1)}$ pulling the prototype for $X_{(2)}$ along in the clustering process.

The scenario in the last paragraph is illustrated with simulated data in Fig. 3. Here $X_{(1)}$ and $X_{(2)}$ have the same number of data points ($N_1 = N_2 = 48$), but $X_{(2)}$ contains only part of a circle (16 points) and the rest are noise points (32 points). The underlying circles have the same radius, $r_1^* = r_2^* = 8$, and are centered at (0,0). We apply our modified PCSC algorithm using different values of α , Fig. 3(a)-(d) correspond to $\alpha = 0, 5, 20$, and 400, respectively. Initial

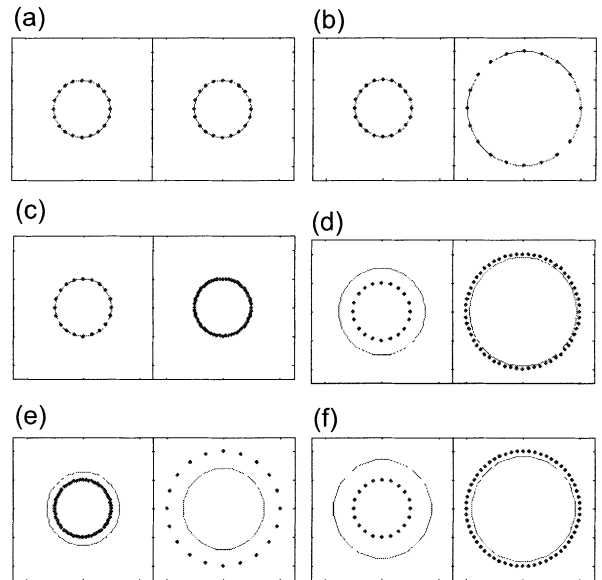


Fig. 2. Simulation example of PCSC with constraint using noiseless data. The small dots are data points to be clustered, and the circles are found prototypes found by our algorithm. Both axes of each small rectangle have ranges of -25 to 25 . Parameters used are: (a) $r_1^* = r_2^* = 10$, $N_1 = N_2 = 20$, $\alpha = 0$; (b) $r_1^* = 10$, $r_2^* = 20$, $N_1 = N_2 = 20$, $\alpha = 0$; (c) $r_1^* = r_2^* = 10$, $N_1 = 20$, $N_2 = 60$, $\alpha = 20$; (d) $r_1^* = 10$, $r_2^* = 20$, $N_1 = 20$, $N_2 = 60$, $\alpha = 20$; (e) $r_1^* = 10$, $r_2^* = 20$, $N_1 = 60$, $N_2 = 20$, $\alpha = 100$; (f) $r_1^* = 10$, $r_2^* = 20$, $N_1 = 20$, $N_2 = 60$, $\alpha = 100$. The left and right rectangles of each of (a)-(d) correspond to parameters with subscripts 1 and 2, respectively.

prototypes are set to be the underlying circles so that no error is caused by bad initialization. The value of η is set to $(r_1^*)^4/2 \approx 2000$. The solid circles are the final prototypes after 10 iterations of our algorithm, and the dashed circles are the underlying circles overlapped with $X_{(2)}$ for comparison. The actual centers and radii of the prototypes found are also listed in Table 1. It is evident that, by increasing α , we are able to better identify the underlying circles in the noisy data set ($X_{(2)}$) through clustering. For the two larger α values, the prototypes found for cluster #2 are so close to the underlying circles that they are almost inseparable.

To more systematically investigate the power of the constraint term in locating the correct underlying circles for this scenario, we repeat for 100 times the same experiment as in Fig. 3 with the same underlying circles but different, randomly generated noise points. Furthermore, the initial prototypes have randomly selected centers within $r_1^*/4 \approx 2$ from (0,0) and radii of between 8 ± 2 . We include the variation in initialization to make the experiment more realistic. The statistics is listed in Table 2. The quantity "radius error" is the absolute radius difference between the prototypes found and the underlying circles. The quantity "deviation" represents the maximum separation between the final prototype and the underlying circle and is used as a measure of how accurately the clustering algorithm finds the underlying circle. Cluster #1, which corresponds to the noiseless set of data points ($X_{(1)}$), remains very close to the

underlying circle for all α values. On the other hand, for cluster #2, which corresponds to the noisy set of data points ($X_{(2)}$), both deviation and radius error are significantly reduced when α is increased. Even $\alpha=1$, which is much smaller than both N_1 and N_2 , results in significantly more accurate prototypes than when $\alpha=0$.

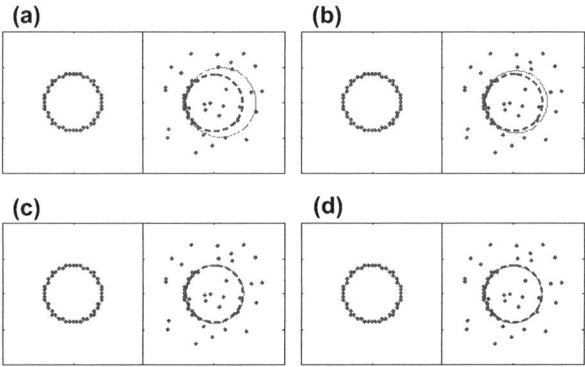


Fig. 3. Simulation example of PCSC with constraint using one set of noiseless data and one set of noisy data. The small dots are data points to be clustered, and the circles are found prototypes found by our algorithm. Both axes of each small rectangle have ranges of -20 to 20 . (a)-(d) correspond to $\alpha=0, 5, 20$, and 400 , respectively. The left and right rectangles of each of (a)-(d) correspond to parameters with subscripts 1 and 2, respectively.

TABLE 1
PROTOTYPES FOUND IN FIG. 3

Plot	α	Cluster #1 (left)		Cluster #2 (right)	
		Center	Radius	Center	Radius
(a)	0	(0.00, 0.00)	8.01	(2.10, 0.12)	9.85
(b)	5	(0.00, 0.00)	8.09	(0.15, 0.05)	8.79
(c)	20	(0.00, 0.00)	8.07	(0.33, 0.04)	8.20
(d)	400	(0.00, 0.00)	8.05	(0.21, 0.02)	8.05

TABLE 2
PROTOTYPE ACCURACY

α	Cluster #1		Cluster #2	
	Deviation	Radius Error	Deviation	Radius Error
0	0.01 ± 0.00	0.01 ± 0.00	3.38 ± 9.95	1.41 ± 4.35
1	0.02 ± 0.02	0.02 ± 0.02	2.16 ± 1.40	0.88 ± 1.05
5	0.06 ± 0.04	0.06 ± 0.06	1.51 ± 0.92	0.55 ± 0.67
20	0.07 ± 0.06	0.07 ± 0.09	0.92 ± 0.49	0.25 ± 0.30
100	0.08 ± 0.06	0.08 ± 0.09	0.71 ± 0.36	0.13 ± 0.15
400	0.08 ± 0.06	0.08 ± 0.10	0.65 ± 0.33	0.09 ± 0.11

Accuracy of prototypes found by our PCSC algorithm using one set of noiseless data and one set of noisy data.

TABLE 3
PROTOTYPE ACCURACY

α	Deviation	Radius Error
0	1.60 ± 2.66	0.59 ± 0.88
1	1.32 ± 1.44	0.48 ± 0.60
5	1.12 ± 0.82	0.42 ± 0.49
20	1.04 ± 0.77	0.39 ± 0.45
100	1.03 ± 0.69	0.40 ± 0.45
400	1.03 ± 0.69	0.39 ± 0.45

Accuracy of prototypes found by our PCSC algorithm using two sets of noisy data.

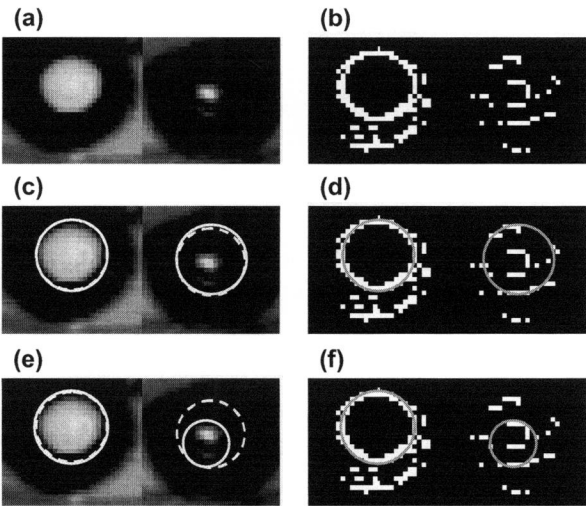


Fig. 4. Processing of eye images in with the PCSC algorithm with constraint. (a) The eye images. (b) The detected potential pupil boundary pixels of (a). (c) and (d) are the same as (a) and (b), with solid and dashed circles indicating pupil boundary locations obtained with our algorithm and by manual labeling, respectively. (e) and (f) are the same as (c) and (d), except that the original PCSC is used.

It is also interesting to investigate the performance of our algorithm when both $X_{(1)}$ and $X_{(2)}$ are noisy. For this purpose we repeat the experiment with both $X_{(1)}$ and $X_{(2)}$ consisting of 24 points along a partial circle and 24 noise points. The statistics of 100 randomized trials are listed in Table 3; the parameters and initialization conditions are the same as those used in generating Table 2. While there is still some improvement in accuracy here with increased α , this improvement is evidently less dramatic as compared to the results in Table 2 even with very large α . The reduction in deviation and radius error seems to level off at larger α , indicating that we can not achieve better accuracy by increasing α .

IV. EXAMPLE FOR REAL IMAGE DATA

After all the experiments with simulated data, we show an example of the application of our modified PCSC algorithm on real data. Here the goal is to simultaneously find the pupils, which are circles, from a pair of eye images obtained in a screening test [12]. The procedure to extract the pupil boundary pixels from the images is given in [13].

Fig. 4 shows a case where the contrast between the pupils and surrounding regions is very low in one eye. Fig. 4(a) is the original eye images and Fig. 4(b) shows the extracted boundary pixels, which serve as the data points in shell clustering. In Fig. 4(c) and 4(d) the circles (pupil boundaries) obtained through the original PCSC are overlapped with the images in Fig. 4(a) and 4(b). The solid circles indicating the pupil boundaries found by our algorithm and the dashed circles indicating manually labeled pupil boundaries. The boundary of the darker pupil is not correctly located because

the algorithm is affected by data points that do not fall on the pupil boundary. The parameters used for PCSC here are $m=1.5$ and $\eta=2000$. In Fig. 4(e) and 4(f) we plot the clustering results with the modified PCSC using $\alpha=100$. We can see that the new algorithm gives much more accurate pupil boundary locations.

V. CONCLUSION

We present in this paper the use of extra cost function terms as a method for incorporating additional desired properties of cluster prototypes. The focus of our analysis is on the locating circular prototypes with similar radii, although the general approach certainly is not limited to this case. The effect of the constraint strength factor α , as analyzed in sections II and III, should be applicable to other cases with constraints on relations among clusters. We believe that this approach can enhance the applicability of clustering in various applications. Possible directions of future research include the investigation regarding other types of constraints, the effect of adjusting α between iterations, and the effect on overlapped clusters.

ACKNOWLEDGMENT

The authors would like to thank Dr. Gerhard Cibis of Children's Mercy Hospital, Kansas City, Missouri, USA, for fruitful collaboration and the use of image data.

REFERENCES

- [1] R. Krishnapuram, H. Frigui, and O. Nasraoui, "A Fuzzy Clustering Algorithm to Detect Planar and Quadric Shapes", *Proc. NAFIPS*, 1992, pp. 59-68.
- [2] R. Krishnapuram, H. Frigui, and O. Nasraoui, "Fuzzy and Possibilistic Shell Clustering Algorithms and Their Application to Boundary Detection and Surface Approximation - Part I", *IEEE Trans. Fuzzy Systems*, vol. 3, pp. 29-43, 1995.
- [3] R. Krishnapuram, H. Frigui, and O. Nasraoui, "Fuzzy and Possibilistic Shell Clustering Algorithms and Their Application to Boundary Detection and Surface Approximation - Part II", *IEEE Trans. Fuzzy Systems*, vol. 3, pp. 44-60, 1995.
- [4] R. N. Dave, "Fuzzy Shell-Clustering and Application to Circle Detection in Digital Images", *Int. J. Gen. Syst.*, vol. 16, pp. 343-355, 1990.
- [5] F. Hoepfner, "Fuzzy shell clustering algorithms in image processing: fuzzy c-rectangular and 2-rectangular shells", *IEEE Trans. Fuzzy Systems*, vol. 5, pp. 599-613, 1997.
- [6] X.-B. Gao, W.-X. Xie, J.-Z. Liu, and J. Li, "Template based fuzzy c-shells clustering algorithm and its fast implementation", *Proc. IEEE Int'l Conf. Signal Processing*, 1996, pp. 1269-1272.
- [7] I. Gath and D. Hoory, "Detection of Elliptic Shells Using Fuzzy Clustering: Application to MRI Images", *Proc. ICPR*, 1994, pp. 251-255.
- [8] A. W. C. Liew and H. Yan, "An adaptive spatial fuzzy clustering algorithm for 3-D MR image segmentation", *IEEE Trans. Medical Imaging*, vol. 22, pp. 1063-1075, 2003.
- [9] R. C. Gonzales and R.E. Woods, *Digital Image Processing* (2nd Ed.), Prentice Hall, 2002.
- [10] R. Krishnapuram and J. M. Keller, "A Possibilistic Approach to Clustering", *IEEE Trans. Fuzzy Systems*, vol. 1, pp. 98-110, 1993.
- [11] J.C. Bezdek and R.J. Hathaway, "Numerical convergence and interpretation of the fuzzy c-shells clustering algorithm", *IEEE Trans. Neural Networks*, vol. 3, pp. 787-793, 1992.
- [12] G. W. Cibis, "Video Vision Development Assessment (VVDA): Combining the Brückner Test with Eccentric Photorefractive for Dynamic Identification of Amblyogenic Factors", *Trans. Am. Ophthalmol. Soc.*, vol. 84, pp. 643-685, 1994.
- [13] T. Wang, "Eye Location and Fixation Estimation Techniques for Automated Video Vision Development Assessment", Master's Thesis, Dept. Comp. Eng. Comp. Sci., University of Missouri-Columbia, Columbia, MO, 2002.